system control unit 1. The syndrome calculator 5 also provides the system control unit 1 with the error-containing code word signal 23 indicating the code word from which an error has been detected.

At the same time, the error detector 7 executes an error detecting process. Prior to the error detection, the mid-term results of the EDCs in the preceding code words stored in the corresponding one of the mid-term result registers 801-816 are reloaded. If the syndrome is zero when the transfer of each code word is over, the mid-term results of the EDCs are stored in the corresponding mid-term result register again. When the syndrome is not zero, on the other hand, the mid-term results of the EDCs in the preceding code words are maintained, without updating the contents of the corresponding mid-term result register.

In the first code word (the first line of the horizontal direction), the mid-term result register is initialized because it contains no mid-term results. When the first detection of an error-containing code in the sector in process is informed by the error-containing code detection signal 22, the contents of the corresponding mid-term result register are not updated, and the subsequent code words are not subjected to error detection.

Step (e-5): the error corrector 6 receives data read from the buffer memory 4 by means of the error corrector access signal 18 outputted by the bus control unit 3, corrects an error in the code, and transmits the access request signal 17 to the bus control unit 3 to request writing of the error-corrected data to the buffer memory 4 again.

Step (e-6): after putting the data bus 11 in commission, the bus control unit 3 reads the error-corrected data from the error corrector 6 and writes them to the buffer memory 4. When error correction for one sector is complete, the error corrector 6 outputs the correction completion signal 19 to the system control unit 1.

Step (e-7): the system control unit 1 outputs the DMA command 12 to the DMA control unit 2 in order to check to see that the error-corrected data contain no error in the sector in process, and provides instructions for data transfer from the buffer memory 4 to the error corrector 7. This data transfer involves data from the code word indicated by the error-containing code word signal 23 outputted together with the error-containing code detection signal 22 outputted first in the sector in process by the syndrome calculator 5 at step (b-4) up to and including the final code word in the sector. This is within the re-calculation range of an EDC shown in Figure 13, which eliminates the need for transfer of data in the valid range of the mid-term results of an EDC in each sector.

Step (e-8): the DMA control unit 2 outputs the DMA request 13 to the bus control unit 3 so as to request the data transfer from the buffer memory 4 to the error detector 7.

Step (e-9): after putting the data bus 11 in commission, the bus control unit 3 outputs the buffer memory access signal 14 to the buffer memory 4 to read the data therefrom. Then, the bus control unit 3 outputs the error detector data supply signal 2 to the error detector 7 so as to supply the data read from the buffer memory 4.

Step (e-10): the error detector 7 executes error detection for the transferred subsequent data, by using the mid-term results of error detection stored in the corresponding mid-term result register. The error detector 7 then transmits the error detection signal 21 to the system control unit 1 so as to inform whether an error has been detected or not.

Thus, error correction and error detection for one sector is complete. The horizontal error correction for one ECC block is completed by repeating these steps for 16 sectors.

In Embodiments 1 and 3 having the single mid-term result register 8, the re-transfer of data in the case where an error has been detected is started from the error-containing code word detected first in one ECC block. In contrast, in the present embodiment having the 16 mid-term result registers corresponding to the 16 sectors, it becomes possible to start the re-transfer of data from the error-containing code word detected first in each sector. This further reduces the time required for error correction and the power consumption.

The present embodiment has 16 mid-term result registers to be provided to the 16 sectors in one ECC block. Instead, one ECC block can be divided into regions each composed of several sectors (four, for example), and the same number of (four) mid-term result registers can be provided. Thus, while reducing the number of mid-term result registers, the data transfer when an error has been detected can be started from the error-containing code word first detected in the divided regions. This reduces the time required for error correction and the power consumption in the same manner as in the present embodiment.

(Embodiment 6)

In the present embodiment, error correction is performed concurrently in three different ECC blocks by pipeline processing.